EXPERIMENT REPORT

Student Name: Yasaman Mohammadi
Project Name: advmla-2023-spring

Date:10/11/2023
Deliverables:LightGBM.ipynb

1.EXPERIMENT BACKGROUND

1.a. Business Objective

This project aims to build a Streamlit app that helps users in the USA estimate local travel airfare. Users input trip details, and the app uses four machine-learning models to predict flight fares. Accurate results improve user trust, save money, and time, and benefit the business. Incorrect results can erode trust, lead to suboptimal decisions, and harm the business's reputation.

1.b. Hypothesis

Hypothesis:

Booking a trip well in advance influences the ticket price, with earlier bookings resulting in lower prices. This is due to dynamic pricing strategies and potential cost savings when airlines can plan efficiently.

1.c. Experiment Objective

The outcome of accurately estimating ticket prices in the experiment would provide strong evidence that users can rely on this site as their sole source for planning their trips. Here are the possible scenarios related to this outcome:

1. **Accurate Estimation Confirmed:**
   If the experiment demonstrates that the models can accurately estimate ticket prices, users can confidently rely on this site for planning their trips. They will have a trustworthy tool for budgeting their travel expenses.

2. **Accurate Estimation Not Confirmed:**
   If the models do not accurately estimate ticket prices, users may still use the site as a reference but not rely solely on it. They might continue to check other sources or use the site as a supplementary tool to get a general idea of prices.
3. **Partial Accuracy Achieved:**
   In some cases, the models might accurately estimate prices for specific routes or scenarios but not for others. Users may rely on the site for specific trips but not for all. The reliability may vary based on the specific flight details.

Ultimately, the success of users relying solely on the site for trip planning depends on the accuracy and consistency of the price estimations provided by the models. If the models consistently deliver precise estimates, users are more likely to rely on the site as their primary source for planning trips.

2.EXPERIMENT DETAILS

2.a. Data Preparation

Table 1- the steps taken for preparation of the data for previous experiment.

	the steps taken for preparing the data	Explanation
1	Data collection	The process of constructing a machine learning model commences with gathering data from diverse sources. In this instance, the data was obtained from Canvas UTS, consisting of four separate CSV files. Then they merged.
3	Explore Dataset	A general understanding is derived from the data in this step.
4	Data cleaning	Data was pre-processed to remove any irrelevant, missing, or corrupted data points and any discrepancies in the data.
5	Data visualization	Visual understanding of data distribution is gained by plotting different charts in Python.
6	Feature engineering	New features added in this step and some features were removed.
7	Splitting the dataset	The dataset was divided into two subsets: a training set, utilized to train the model, and a test set, employed to assess the model's final performance. The test size parameter was set to 20%, indicating that the last 20% of the dataset was designated as the test data.

Missing Values and Percentage:

	Missing Values	Missing Percentage
totalTravelDistance	959619	7.097774
segmentsEquipmentDescription	262676	1.942870

Figure2-Missing value

2.b. Feature Engineering

"*daysInadvance":*
Examines how many days exist between the date of the flight's search and the actual flight date, (entails the sum of the no. days between the flightdate and searchDate) - included, as airlines often adjust pricing depending on how early or late a booking takes place.

"SearchDate":
following extraction of the day_of_the_week , month , and year from flightDate, values were then converted into an integer format.

Flight_Date:
Following extraction of the day_of_the_week , month , and year from searchDate, values were then converted into an integer format.

"Departure_time":
Extracted from segmentsDepartureTimeRaw. Originally formatted as ('2022-05-20T18:58:00.000-07:00||2022-05-21T00:5...'), the 24hr time was isolated and converted to an integer format.

"SegmentsCabinCode":
Extracted from segmentedCabinCode. This feature acts as an intermediate prior to converting cabin differentiations into percentages.

"numSegments":
The count of stops in each trip, derived from segmentsCabinCode.

"isRefundable":
We encoded this feature by converting its boolean values (True and False) to integers (1 and 0), ensuring it could be effectively utilized as numerical input for the model.

"startingAirport":
One-hot encoding was applied to this feature.

"destinationAirport":
One-hot-encoding was performed on this feature as well.

2.c. Modelling

LightGBM is a gradient boosting framework that uses tree-based learning algorithms and is designed for distributed and efficient training, particularly on large datasets. It was adopted for its computational efficiency and capacity to scale without accuracy loss.

- **N estimator**

  This is the number of boosting rounds or trees trained in the LightGBM model. It is a crucial hyperparameter because it determines how many iterations the model will go through to learn from the data. Increasing the value of "n_estimators" generally allows the model to learn more complex patterns in the data, but it also makes the training process slower and may risk overfitting if set too high.

  With grid search, a n estimator of 350 to 650 was selected for this experiment, and 650 was determined to be the best value.

- ***Max depth***

  This parameter, denoting the maximum depth of the tree, is instrumental in managing model overfitting. Through grid search, a range from 5 to 100 was explored in this experiment, and the optimal value determined was 20.

K-fold cross-validation with five folds is employed in the code to evaluate and fine-tune the LightGBM regressor's hyperparameters robustly. This technique ensures that the model's performance is thoroughly assessed across multiple data subsets, reducing the risk of overfitting, and providing a more reliable estimate of its effectiveness. GridSearchCV systematically explores hyperparameter combinations, such as the number of estimators ("n_estimators"), using the "neg_mean_absolute_error" metric, resulting in a well-tuned and generalized model.

3.EXPERIMENT RESULTS

3.a. Technical Performance

An RMSE value of 117.33 was achieved on the test set.For future experiments, it is worth considering additional hyperparameters since we only used two hyperparameters for this experiment . Additionally, incorporating more features may enhance predictive performance further.

3.b. Business Impact

Correct Results (Booking Timing):
If it is confirmed that booking well in advance leads to lower ticket prices, both users and the business benefit. Users trust the app, engage more, and potentially save money. The app can generate revenue and build a strong reputation in the industry.
Incorrect Results (Booking Timing):
Conversely, if the hypothesis is incorrect, users may not save as expected, leading to disappointment and behavior changes. This can reduce trust, engagement, and satisfaction. The app's reputation may suffer, and potential revenue opportunities could be missed. Airlines may also face consequences in terms of early booking patterns.

3.c. Encountered Issues

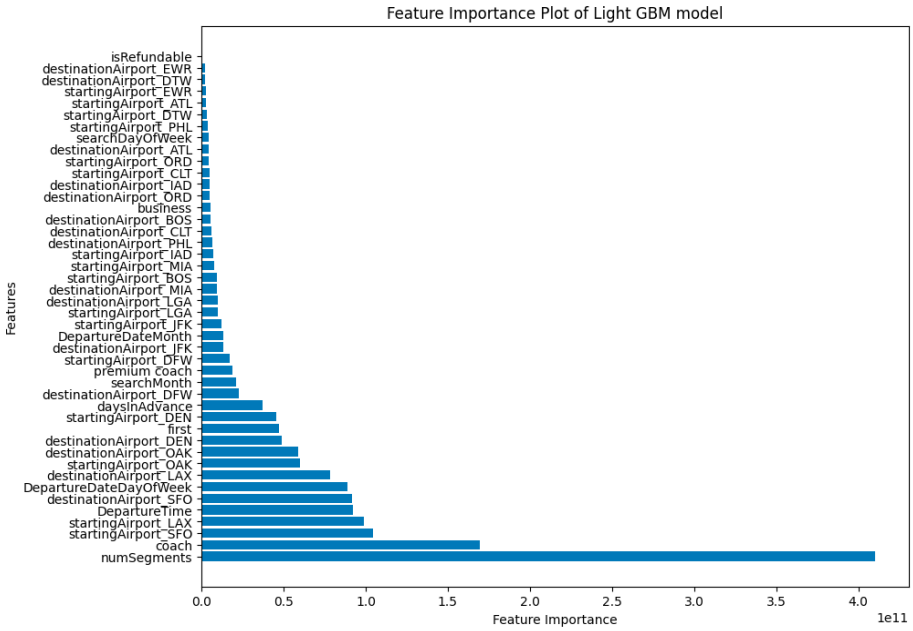
At first, we used categorical feature directly since light gbm can handle this kind of features, but we encountered this error:

```
    raise ValueError('train and valid dataset categorical_feature do not match.')
ValueError: train and valid dataset categorical_feature do not match.
```

So instead, we feature engineered every feature to only have numerical values for the models input.

4.FUTURE EXPERIMENT

4.a. Key Learning



The feature importance plot clearly highlights the significant impact of the number of stops in the trip and the "daysinAdvance" on estimating ticket prices. Additionally, it underscores the influence of the chosen cabin type on ticket pricing.

4.b. Suggestions / Recommendations

In future experiments, we can use additional features to get better accuracy.
To deploy the final solution into production, the user interface must be refined so that predictions are easy to access, and ongoing model adjustments should be monitored in real-time.